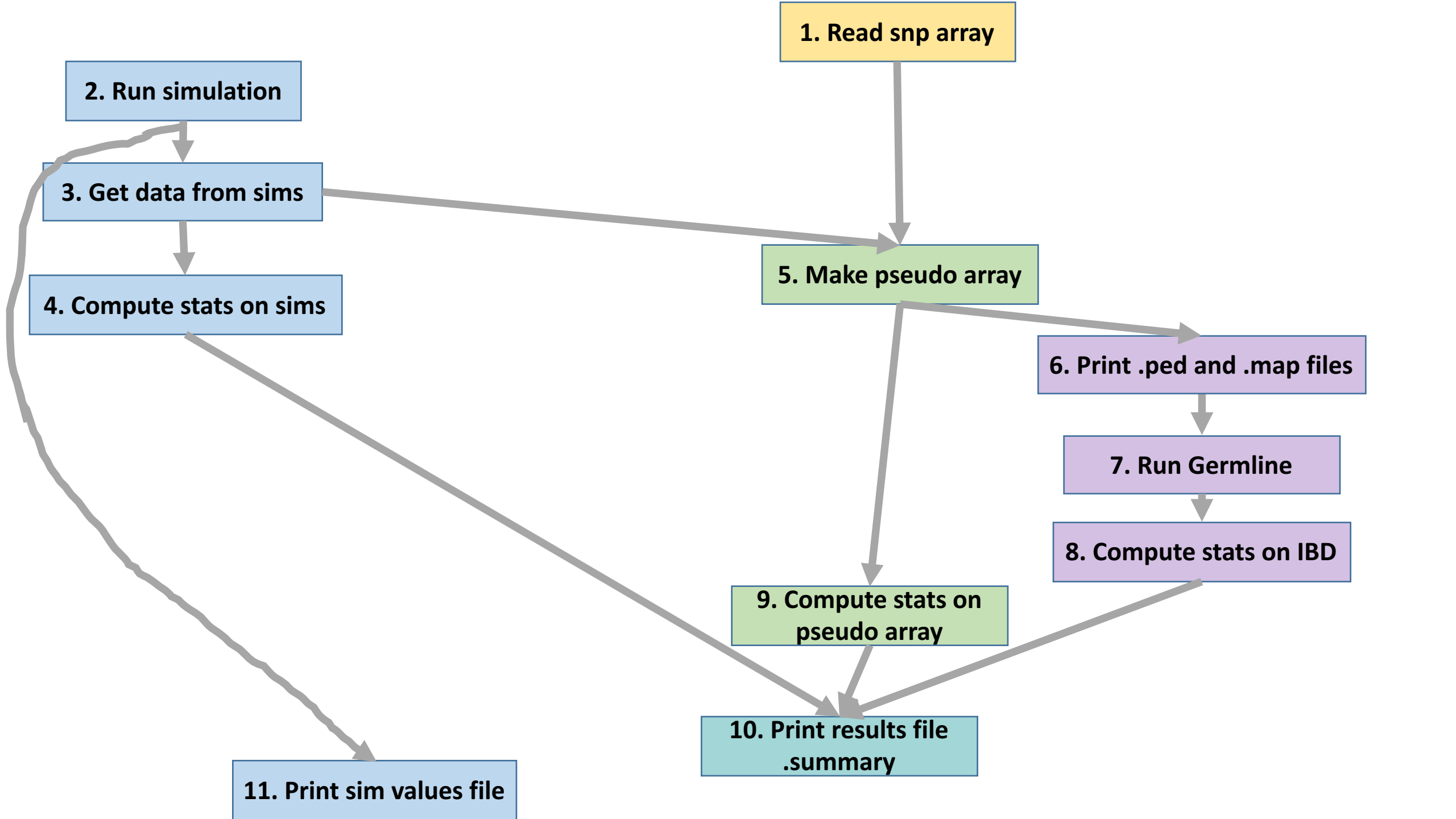
Order of run_sims_AJ_chr1.py excution

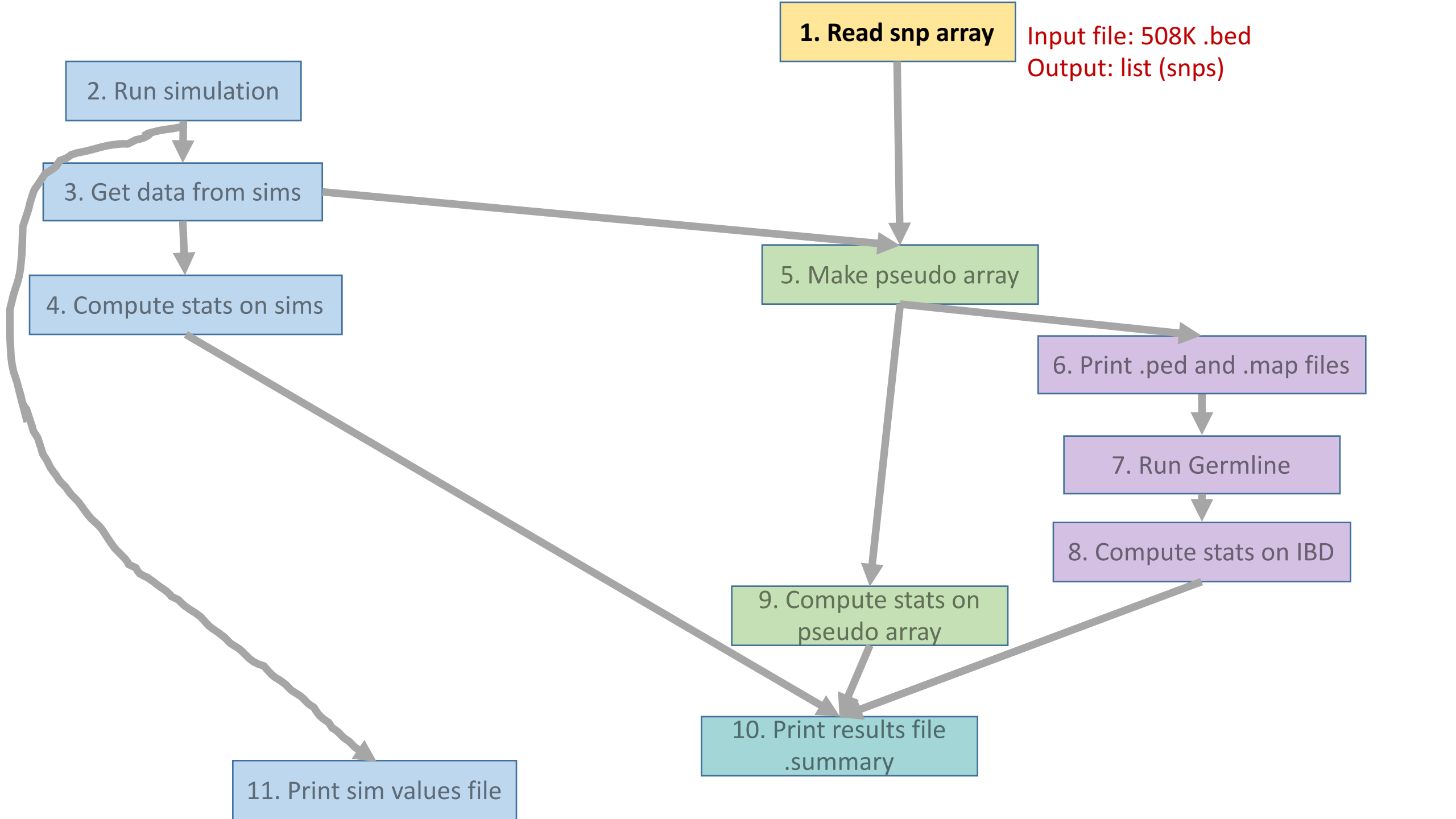
1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print ped and map files
7. Run germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file
11. Print sim values file


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

1. Read snp array

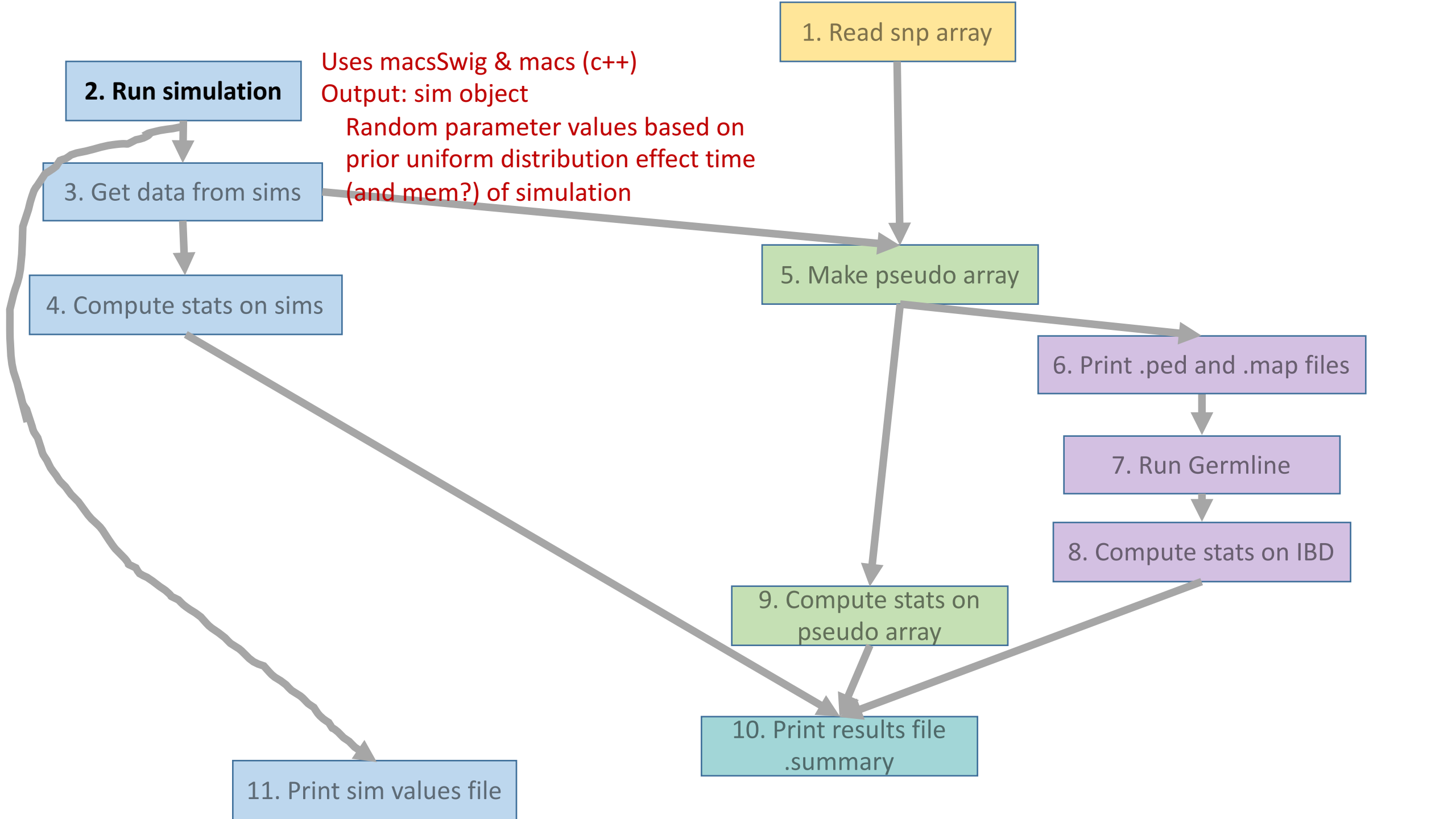
- Input file: ftDNA_hg18_auto_all_uniqSNPS_rmbadsites_pruned_chr1.bed

  Or other .bed snp array file
- 508K
- In run_sims_AJmodel1_chr1.py lines 902-919
- Output: list (snps)


1. Read snp array
Input file: 508K .bed
Output: list (snps)
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

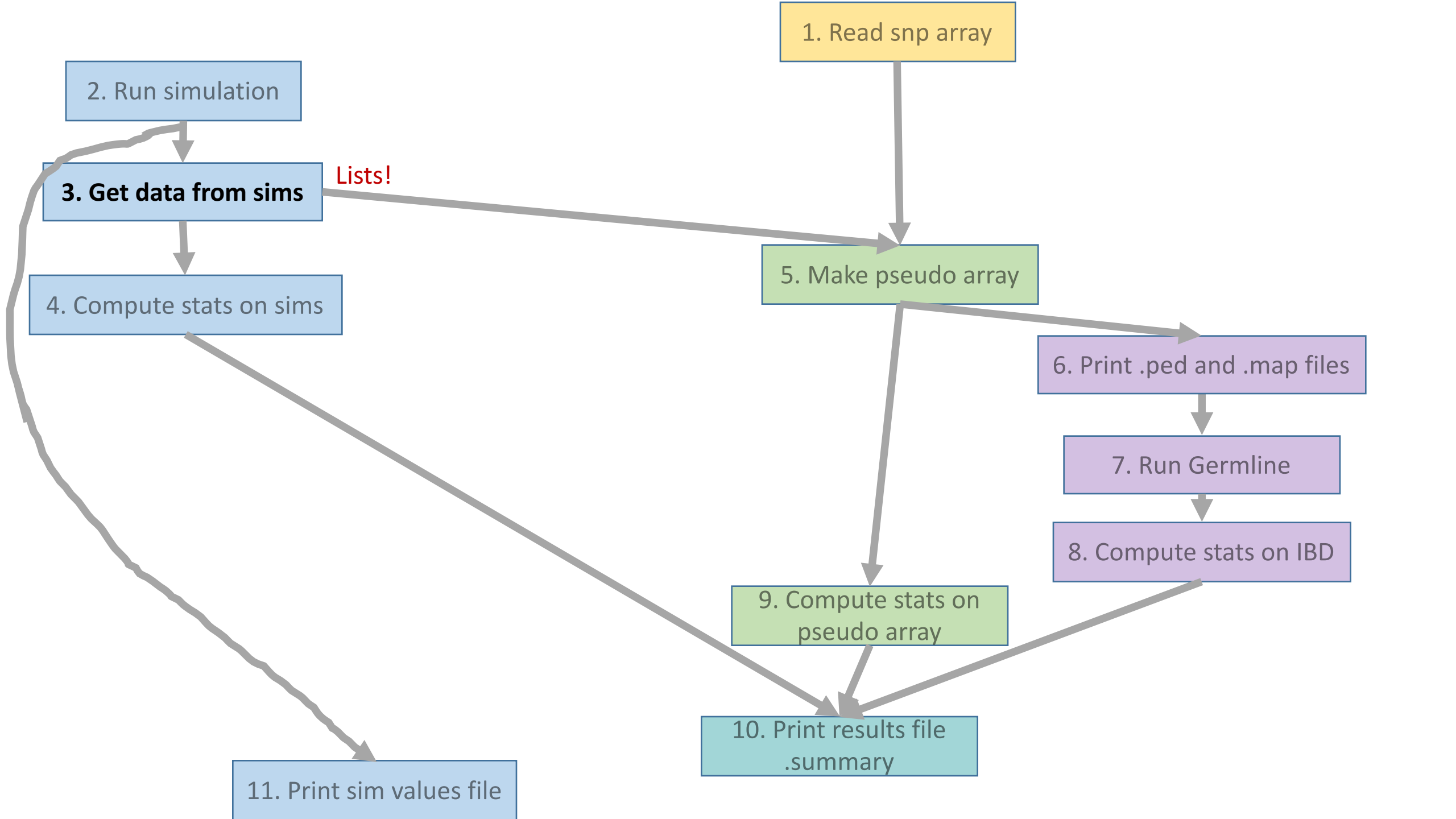
2. Run simulation

- Uses macsSiwg, and macs
  - Macs is a c++ program that performs genome coalescent simulations
    - Efficient
    - https://github.com/gchen98/macs
  - macsSwig is a swig wrapper for macs, which gives the output of macs as a python object (written by August Woerner)
    - Outputs (bit?) object sim
    - macsSwig is run and calls macs on line 939, which calls the function run_sim
      - run_sim (lines 721-811) takes in parameters, case, length, chr_number and defines macs argument as a string
        - Uses function param_sim_asc (lines 501-719), which defines parameters based on given prior distribution and chooses model case
          - Parameters will effect the time (and mem?) of simulation – large N & T parameters = longer


1. Read snp array
2. Run simulation
Uses macsSwig & macs (c++)
Output: sim object
Random parameter values based on prior uniform distribution effect time (and mem?) of simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

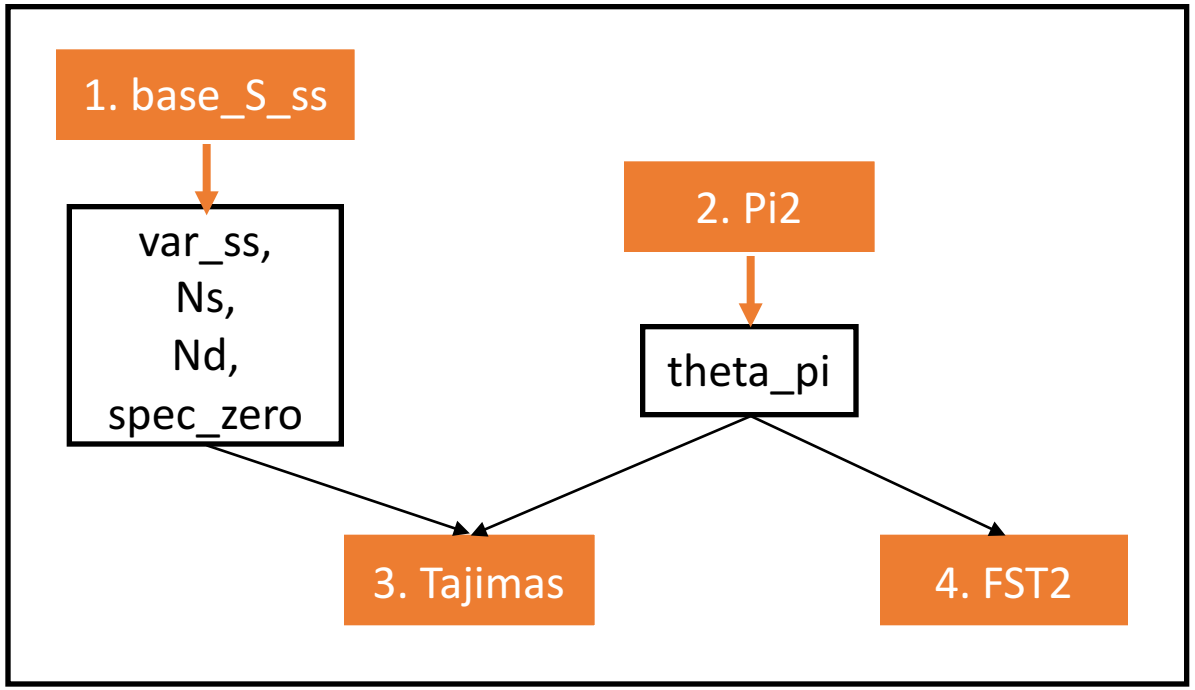
Get data from sims

- Many many lists of all the data and for subsets of the data (each population)
  - pos
  - alleles
  - Talleles
  - seq
  - panel
- Lines 951-1032

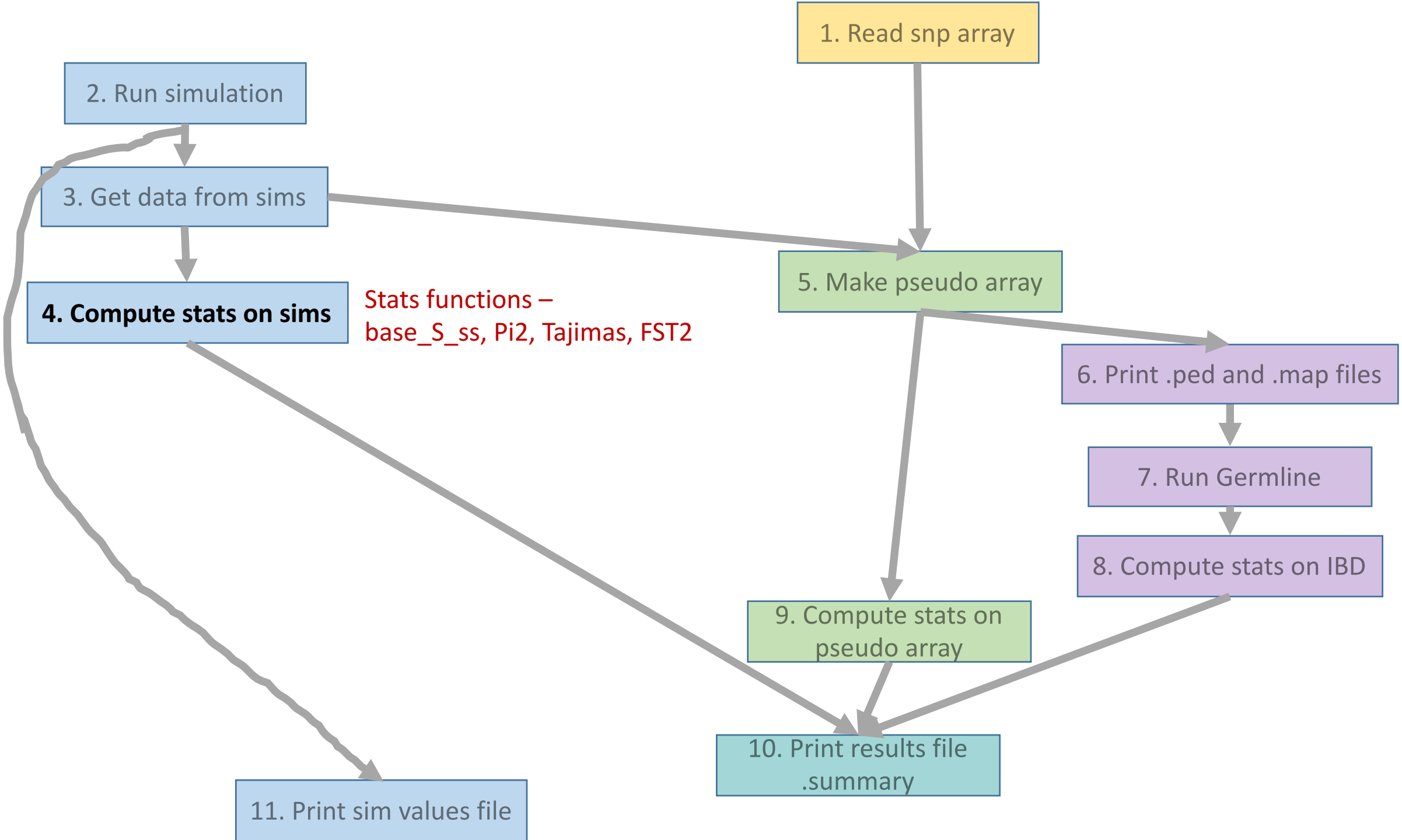

1. Read snp array
2. Run simulation
3. Get data from sims
Lists!
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

Compute stats on sims

- Use stats functions written by Krishna and Consuelo on panel populations
  - base_S_ss (lines 32-64), Pi2 (lines 200-206), Tajimas (lines 209-239), FST2 (lines-341-364)

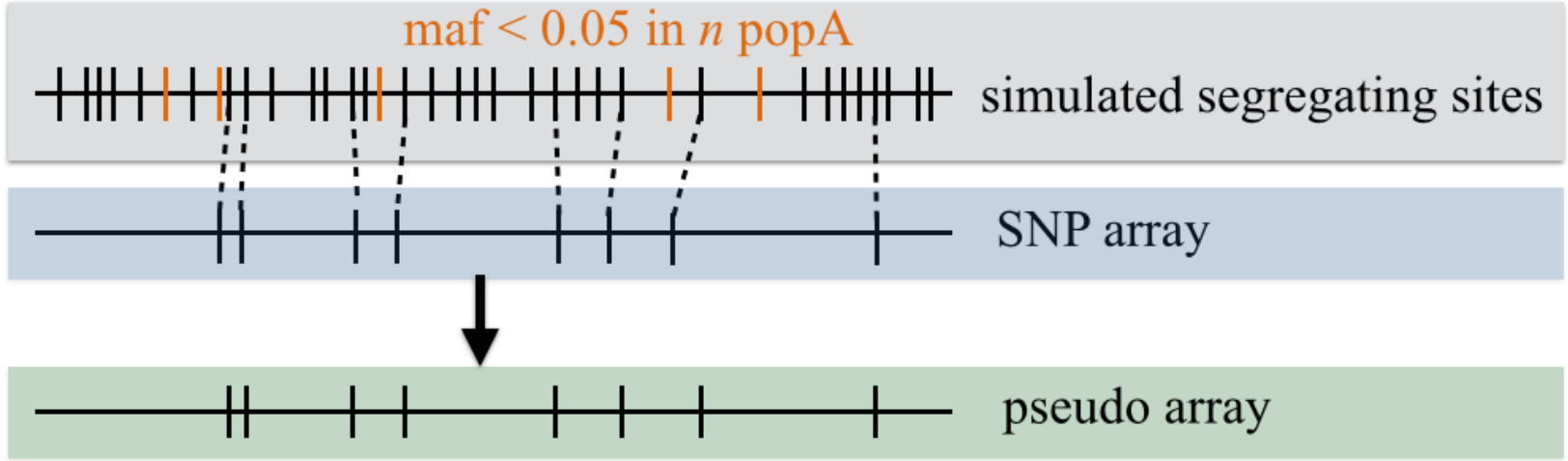


*Do not want to mess these up

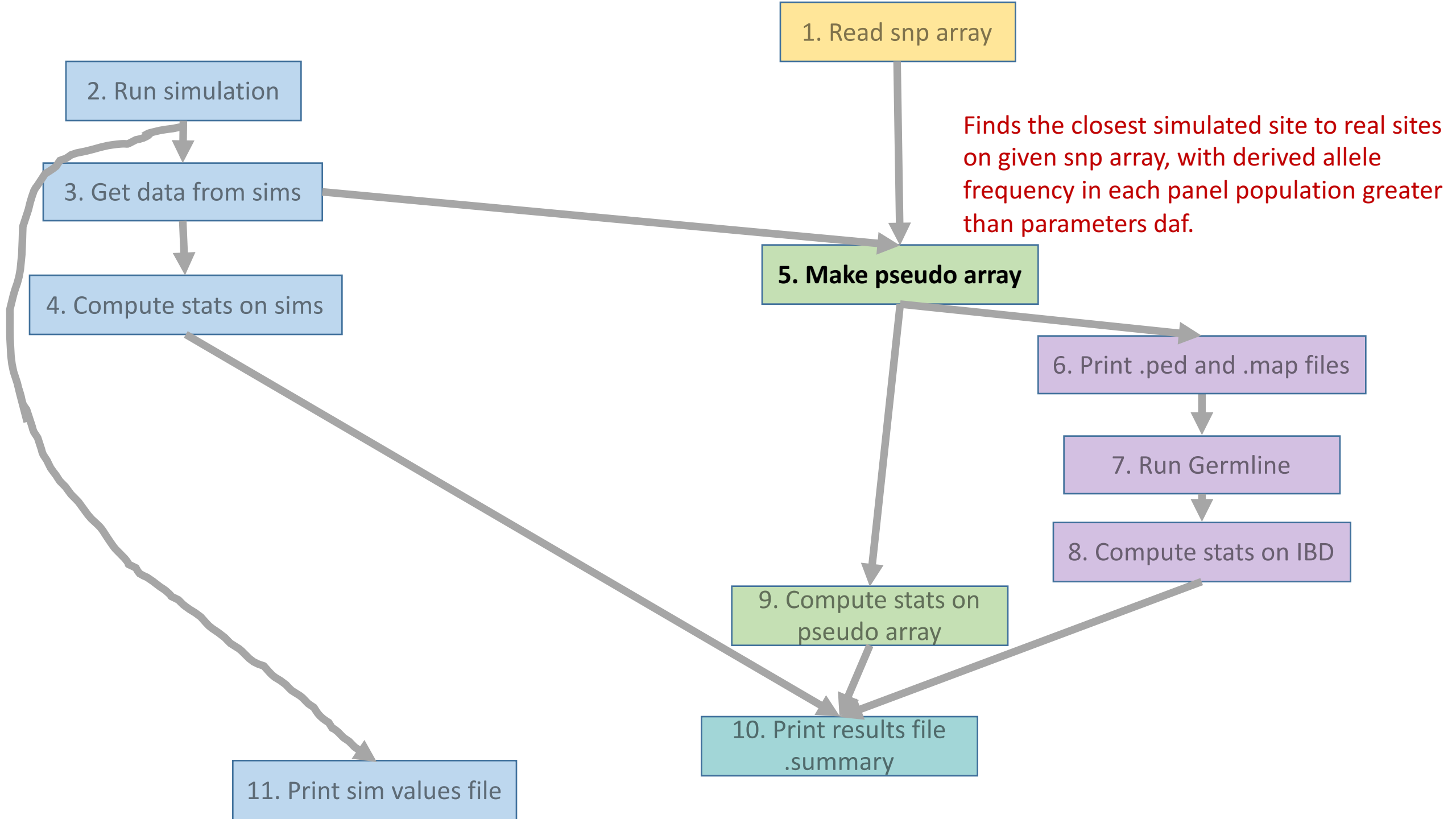

1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
Stats functions –
base_S_ss, Pi2, Tajimas, FST2
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

Make pseudo array

- Uses Consuelo's code (based off my original function find)
- Lines 1100-1208
- Uses function find2
- Finds the closest simulated site to real sites on given snp array, with derived allele frequency in each panel population greater than parameters daf.
- Outputs lists

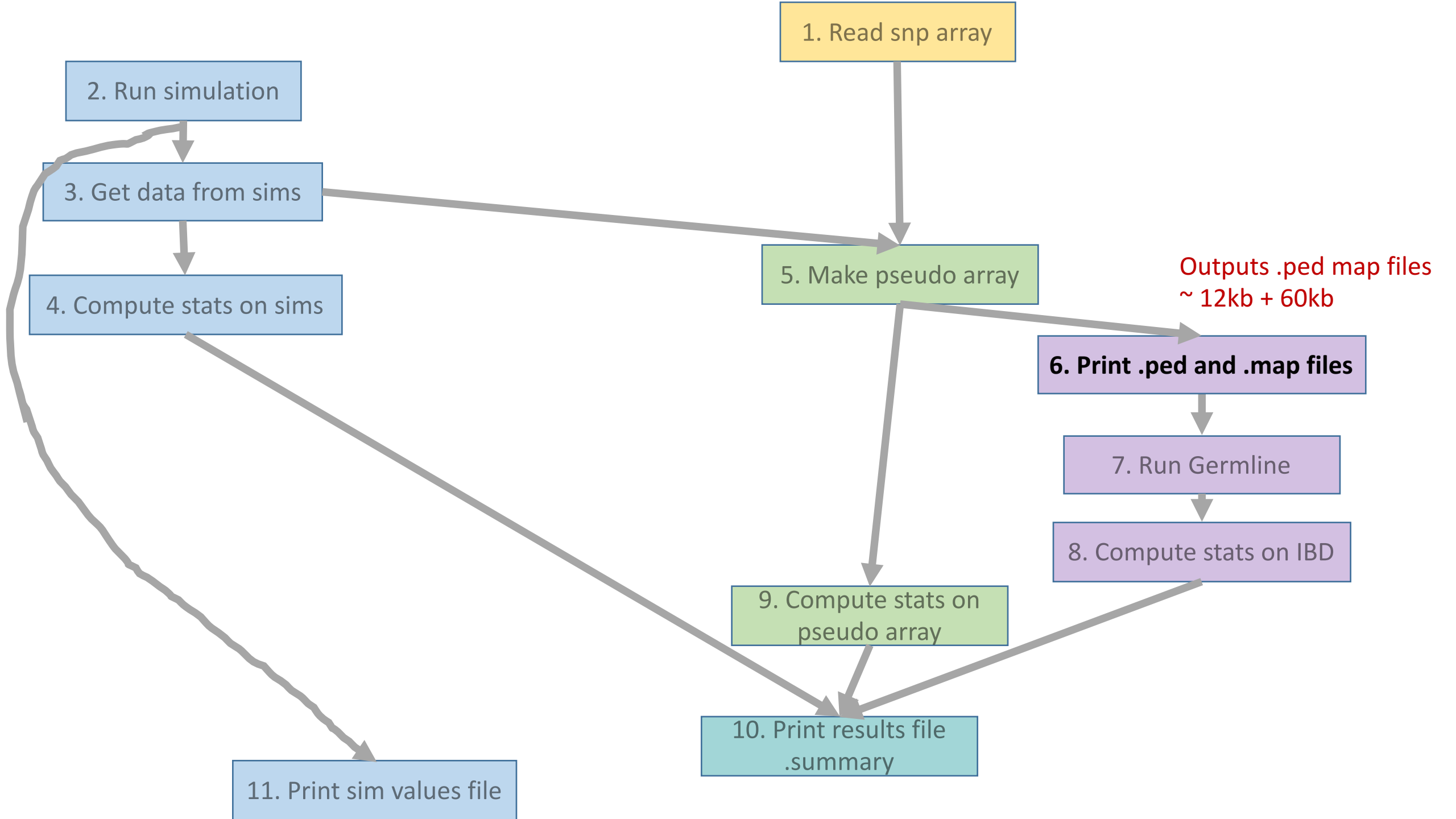


*My original function is not designed to deal with multiple populations in a discovery panel and it does not deal with duplicates.


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
Finds the closest simulated site to real sites on given snp array, with derived allele frequency in each panel population greater than parameters daf.
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

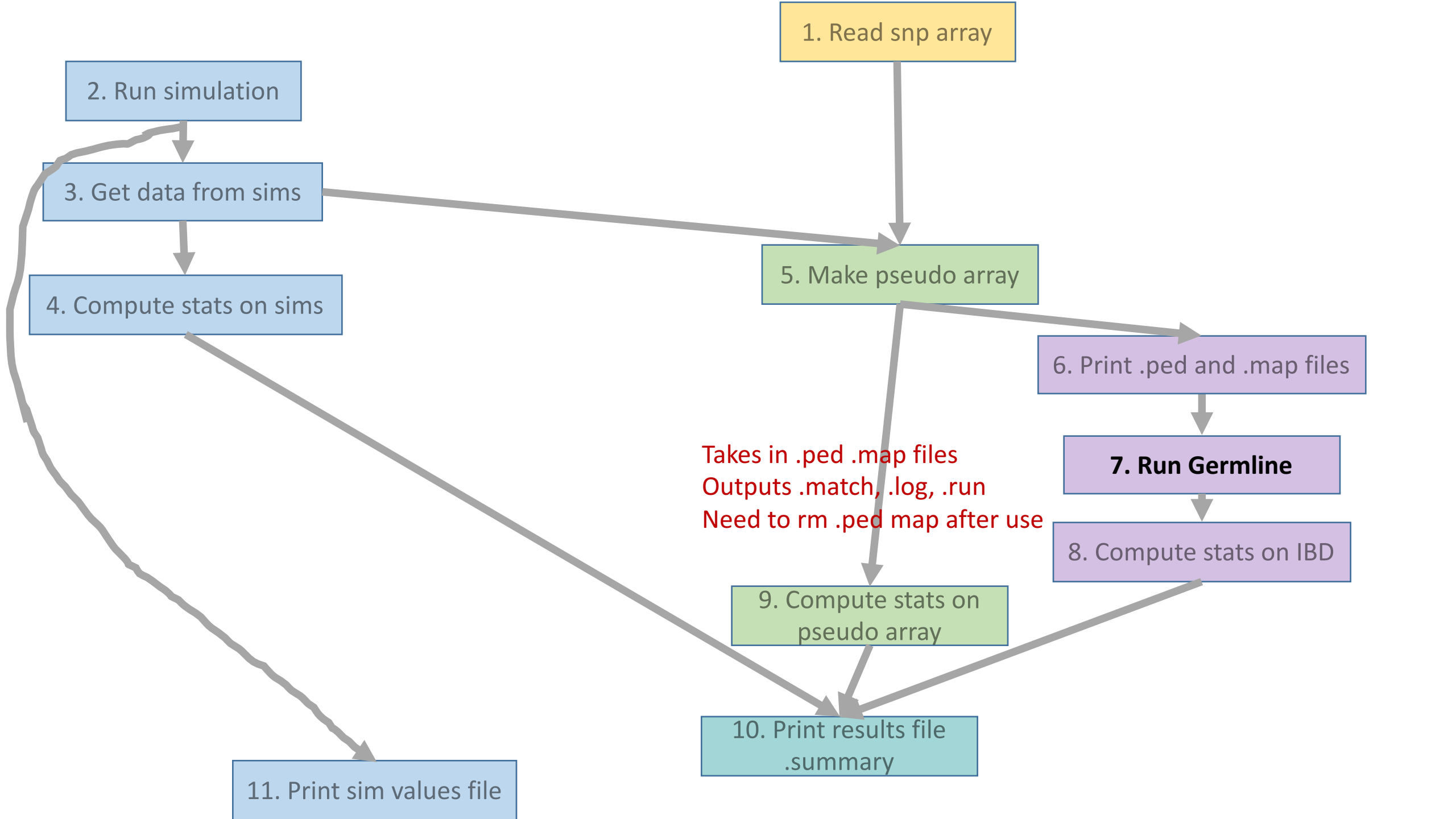
Print ped and map files

- Reads in pseudo array lists and prints in correct .ped .map format
- These files are only needed for Germline
- .ped ~60K
- .map ~12K


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
Outputs .ped map files
~ 12kb + 60kb
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

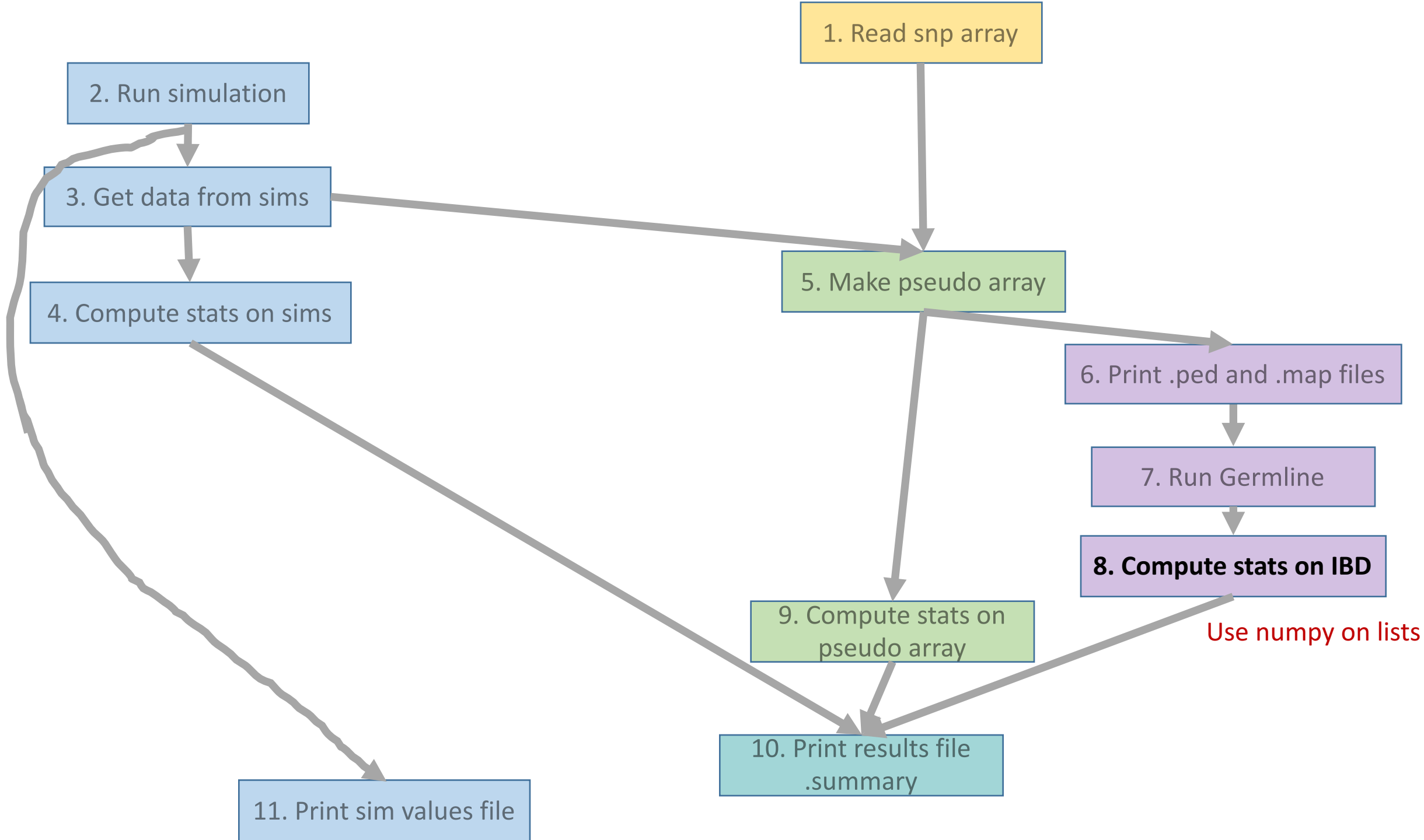
Run Germline

- Uses Popen to run Germline (c++) (lines 1349-1358)
  - https://github.com/sgusev/GERMLINE
  - Takes in .ped .map files
  - Outputs .match (~5M), .log (994b) files
  - Popen is a memory hog!
    - http://stackoverflow.com/questions/1367373/python-subprocess-popen-oserror-errno-12-cannot-allocate-memory/13329386#13329386
    - http://stackoverflow.com/questions/5306075/python-memory-allocation-error-using-subprocess-popen


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
Takes in .ped .map files
Outputs .match, .log, .run
Need to rm .ped map after use
7. Run Germline
8. Compute stats on IBD
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

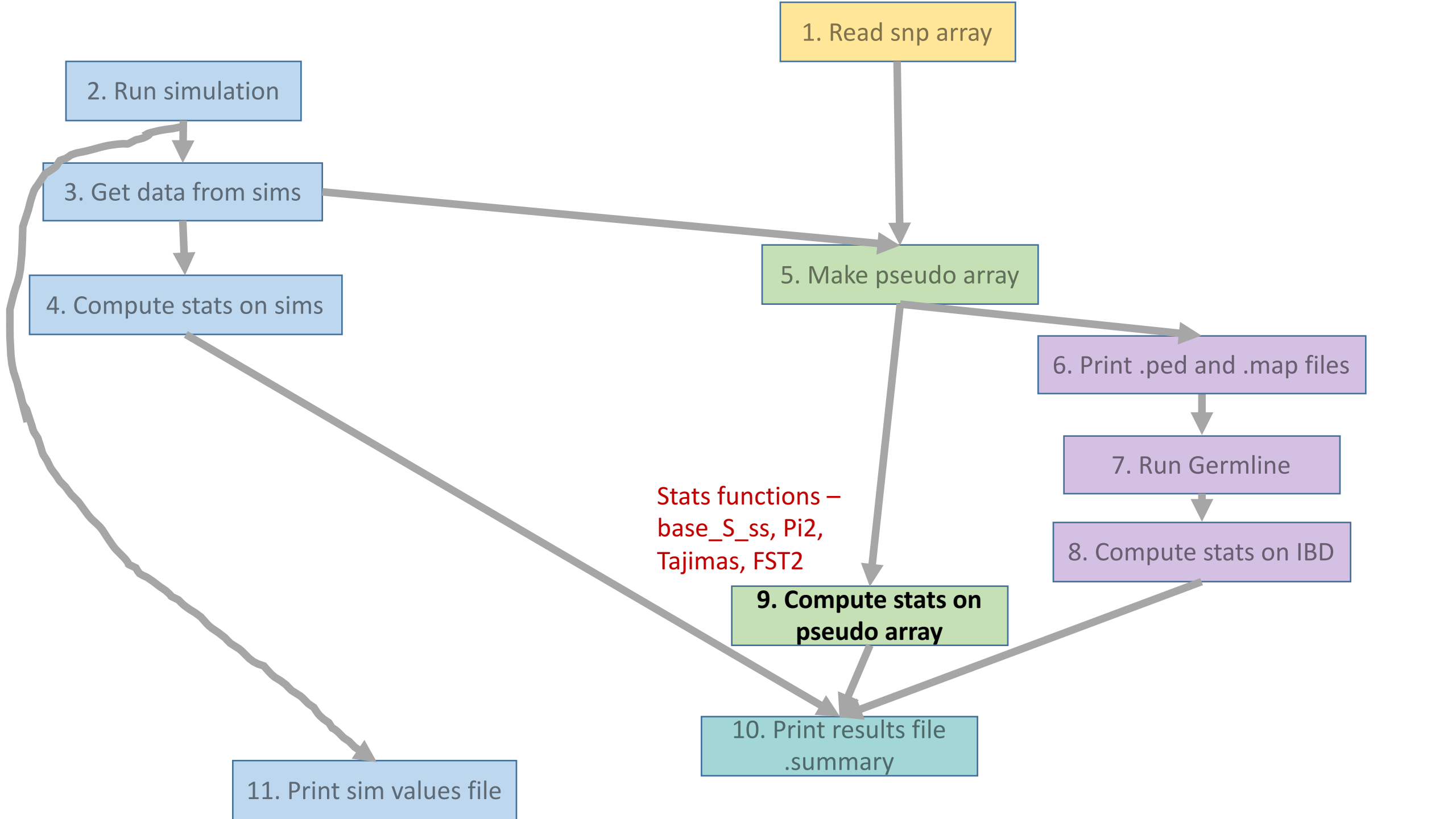
Compute stats on IBD

- Read Germline .match output
- Put each population pair into lists (IBDlengths)
  - IBDlengths varies in size
- Use numpy to compute mean, median, var of lists
- rm .match


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
Use numpy on lists
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file

Compute stats on pseudo array

- Same as compute stats on sims, but on pseudo array


1. Read snp array
2. Run simulation
3. Get data from sims
4. Compute stats on sims
5. Make pseudo array
6. Print .ped and .map files
7. Run Germline
8. Compute stats on IBD
Stats functions – base_S_ss, Pi2, Tajimas, FST2
9. Compute stats on pseudo array
10. Print results file .summary
11. Print sim values file